

Proyecto de análisis de sentimientos con Python

Curso 2023/2024

Objetivo y contexto

si nuestros tweets tienen emoticonos, los tenemos que eliminar
los tweets se tiene que leer frases enteras.

Se pretende un modelo de aprendizaje automático capaz de analizar el sentimiento de frases o textos y aplicarlo para predecir el sentimiento de *tweets*.

El análisis de sentimientos es una rama de la inteligencia artificial que se dedica a analizar las opiniones, emociones y actitudes expresadas en textos escritos, tales como comentarios en redes sociales, reseñas de productos o servicios, artículos de opinión, entre otros. Esta disciplina surge como respuesta a la necesidad de extraer información valiosa a partir de grandes cantidades de datos no estructurados generados por usuarios en internet.

Esta metodología se aplica en diversos campos, desde el marketing y la publicidad hasta la política y la salud mental. En el mundo empresarial, por ejemplo, se utiliza para conocer la satisfacción del cliente y mejorar la calidad de los productos y servicios ofrecidos. En política, puede ayudar a los candidatos a entender cómo se perciben sus discursos y propuestas, mientras que en salud mental puede ser utilizado para detectar signos tempranos de depresión o ansiedad.

Existen herramientas de procesamiento de lenguaje natural, como TextBlob, que permiten realizar tareas de detección de sentimientos.

Durante el desarrollo de este proyecto se aprenderá a usar las herramientas ya existentes para python, como TextBlob, para generar un conjunto de datos de entrenamiento con el que desarrollar vuestro propio modelo de aprendizaje automático capaz de analizar el sentimiento detrás de *tweets*.

Objetivos específicos:

- Generar tu propio conjunto de datos de entrenamiento.
- Usar distintas herramientas de procesamiento de lenguaje natural para depurar el texto recopilado.
- Usar las herramientas ya existentes para el análisis de sentimientos.
- Proponer un modelo de aprendizaje automático que refine la detección de sentimientos, diferenciando entre “Muy feliz”, “Contento”, “Neutro”, “Molesto” y “*Hater*”.
- Usar el modelo desarrollado para clasificar el estado de ánimo de dos personajes públicos según sus *tweets*.

Ejercicio 0: Crear un entorno de trabajo.

Este apartado no tiene puntuación debido a su obviedad. Crea un espacio de trabajo ordenado donde guardar el notebook de Jupyter donde vas a desarrollar este proyecto.

Ejercicio 1: Recopilación de datos. (1p)

Se deben recopilar más de 5000 *tweets* en inglés que puedan tener un sentimiento positivo, negativo o neutro. Procurar mantener en la medida de lo posible una proporción equilibrada. Para la recopilación de los datos se puede hacer de forma manual (no recomendable) o usar un dataset de tweets de la base de datos Kaggle (OJO! Que el texto no esté ya pre-procesado). Los tweets recopilados se deben almacenar en un archivo CSV que se debe leer con pandas. Muestra en el notebook las primeras 5 filas de la tabla leída. [la tabla la tenemos que dejar solo con el texto, extraer una columna del texto.](#)

Ejercicio 2: Limpieza del texto, eliminar las palabras que no aportan información. (2p) [Este ejercicio es el más difícil](#)

Crear una función `limpiar_texto` que toma un dataset como entrada y realiza una serie de pasos de preprocesamiento para limpiar y estructurar el texto de manera que sea más adecuado para tareas de procesamiento de lenguaje natural (NLP).


- Para cada línea de texto del data set, la función debe:
 - Eliminar menciones, hashtags y URLs del texto. Pista: la función `re.sub()` puede ser útil.
 - Convertir el texto en minúscula.
 - Separa las palabras dentro del tweet como elementos de una lista. A este proceso se le llama tokenización, investiga cómo hacerlo con la función `TweetTokenizer()` incluida en NLTK.
 - Para cada una de las palabras en esa lista generada:
 - Eliminar las palabras comunes y poco informativas (conocidas como *stop words*). Usa la recopilación de stop words que ya existe en NLTK (accede a ellas con `stopwords.words()` seleccionando inglés como idioma).
 - Utilizar técnicas de lematización (stemming). Investiga qué son y usa el modelo `SnowballStemmer()` que ya incluye NLTK para stemming.
 - Transformar la lista final en una cadena de caracteres de nuevo. Cada palabra debe estar separada por un espacio y ese texto debe sustituir al original en la tabla del dataset.

Ejercicio3: Etiquetado de datos con herramientas ya existentes. (2p)



El objetivo principal de esta función es asignar la categoría de sentimiento correspondiente usando modelos existentes como TextBlob. TextBlob asigna una puntuación de polaridad al texto, donde valores positivos indican sentimientos positivos y valores negativos indican sentimientos negativos. La intención del proyecto es usar los modelos ya existentes para crear el dataset de entrenamiento necesario para entrenar nuestro propio modelo e intentar que funcione mejor que los ya existentes.

Define una función llamada `clasificador` que toma el dataset con el texto procesado y limpio como entrada. El resultado de este proceso debe almacenarse en un archivo CSV con dos columnas: una para la frase o texto y otra para la etiqueta correspondiente.

- Para cada línea de texto del data set, la función debe:
 - Utilizar el modelo TextBlob para realizar un análisis de sentimiento. Investiga cómo hacerlo.
 - En función de la polaridad calculada, clasificar el sentimiento en categorías específicas. Las categorías deben incluir "Contento", "Muy feliz", "Neutro", "Molesto" y "Hater". 
- Finalmente, guardar los resultados como una columna nueva en el dataset.

Ejercicio4: Codificación de los atributos y objetivos. (1p)

Investiga cómo funciona `CountVectorizer()` e investiga si existe otro codificador que se ajuste mejor a tu tarea. Una vez codificado, separa el conjunto de entrenamiento del conjunto de prueba.

Ejercicio5: Entrenamiento del modelo. (2p)



Entrena un modelo Naive Bayes y otro de tu elección. Investiga sobre qué versiones de cada modelo es más adecuada para la tarea a realizar. Calcula el porcentaje de acierto del modelo entrenado. Si el porcentaje de acierto está por debajo del 70%, vuelve a atrás e intenta mejorar este resultado.

Ejercicio6: Usar el modelo entrenado (2p)

Se deben recopilar los últimos 30 *tweets* de dos personas con influencia en redes sociales de forma manual, alguna conocida por su mala fama como “*hater*” y otra con una valoración social más positiva. Utilizar el modelo desarrollado para predecir el sentimiento de cada *tweet*. Los resultados para cada persona deben ser ilustrados en un pie chart (diagrama de sectores) que muestre el estado de ánimo (el porcentaje de *tweets* clasificados con cada etiqueta).



Documentación y entrega

Deberá entregarse en un jupyter notebook en el que se explique el código desarrollado.